

**Configuration for generating a clock including a delay circuit and method thereof**

Patent Number: ☒ [US2002075047](#)  
Publication date: 2002-06-20  
Inventor(s): TSUKIKAWA YASUHIKO (JP)  
Applicant(s): MITSUBISHI ELECTRIC CORP (US)  
Requested Patent: ☒ [JP2002185313](#)  
Application Number: US20010877027 20010611  
Priority Number(s): JP20000385020 20001219  
IPC Classification: H03L7/06  
EC Classification: [H03L7/081A1](#), [G11C7/22](#)  
Equivalents: ☒ [DE10136163](#)

---

**Abstract**

---

A delay locked loop (DLL) employs a gray code (an alternate code) counter as a delay register. Preventing a carry from arising at more than one bit can minimize skipping of delay time (discontinuous skipping thereof) if a metastable state should occur

---

Data supplied from the esp@cenet database - I2

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2002-185313  
(P2002-185313A)

(43) 公開日 平成14年6月28日 (2002.6.28)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
H 0 3 L 7/081		H 0 3 K 5/14	5 B 0 7 9
G 0 6 F 1/10		23/64	G 5 J 0 0 1
G 1 1 C 11/407		H 0 3 M 7/16	5 J 1 0 6
H 0 3 K 5/00		H 0 3 L 7/08	J 5 M 0 2 4
5/14		G 0 6 F 1/04	3 3 0 A

審査請求 未請求 請求項の数17 O L (全 20 頁) 最終頁に続く

(21) 出願番号 特願2000-385020 (P2000-385020)

(22) 出願日 平成12年12月19日 (2000.12.19)

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 月川 靖彦

東京都千代田区丸の内二丁目2番3号 三

菱電機株式会社内

(74) 代理人 100064746

弁理士 深見 久郎 (外4名)

Fターム(参考) 5B079 CC04 CC14 DD08

5J001 AA11 BB13 BB21 BB23 DD09

5J106 CC21 CC58 DD17 DD24 DD33

5M024 AA21 AA49 BB27 BB35 BB36

DD83 GG02 GG20 JJ02 JJ38

PP01 PP02 PP03 PP07 PP10

(54) 【発明の名称】 デイレイロックドループ、当該ディレイロックドループを含む半導体装置およびクロック同期により動作するシステムのための制御方法

(57) 【要約】

【課題】 安定したクロックを発生するディレイロックドループ、当該ディレイロックドループを含む半導体装置およびクロック同期により動作するシステムのための制御方法を提供する。

【解決手段】 デイレイロックドループ (DLL) において、遅延レジスタとしてグレイコードカウンタ (グレイコード: 交番符号) を用いる。これにより、キャリーを常に1ビットしかたてないようにすることで、仮にメタステーブル状態が起きても遅延時間の飛び (不連続な飛び) を最小化することができる。

4ビットグレイコードの説明

デシマルコード	バイナリコード ADR<3~0>	グレイコード GADR<3~0>	・キャリー
	<3> <2> <1> <0>	<3> <2> <1> <0>	
0	0 0 0 0	0 0 0 0	
1	0 0 0 1	0 0 0 1	
2	0 0 1 0	0 0 1 1	
3	0 0 1 1	0 0 1 0	
4	0 1 0 0	0 1 1 0	
5	0 1 0 1	0 1 1 1	
6	0 1 1 0	0 1 0 1	↓ ダウン方向
7	0 1 1 1	0 1 0 0	
8	1 0 0 0	1 1 0 0	
9	1 0 0 1	1 1 0 1	
10	1 0 1 0	1 1 1 1	
11	1 0 1 1	1 1 1 0	
12	1 1 0 0	1 0 1 0	
13	1 1 0 1	1 0 1 1	
14	1 1 1 0	1 0 0 1	↑ アップ方向
15	1 1 1 1	1 0 0 0	

## 【特許請求の範囲】

【請求項1】 第1クロックを遅延して第2クロックを出力する遅延回路と、

前記第1クロックと前記第2クロックとの位相差を検出する検出器と、

前記検出器の出力に応じて前記遅延回路の遅延量を調整する信号を発生する、グレイコードを用いたグレイコードカウンタとを備える、ディレイロックドループ。

【請求項2】 前記グレイコードカウンタは、

前記グレイコードを格納するグレイコードレジスタと、

前記グレイコードをバイナリコードに変換するバイナリコード変換器と、

前記バイナリコード変換器に格納されるバイナリコードによりアップキャリア信号とダウンキャリア信号とを発生するアップキャリア／ダウンキャリア発生器と、

前記検出器の結果に応じて、前記アップキャリア信号および前記ダウンキャリア信号から、前記グレイコードレジスタにおけるグレイコードを更新するためのキャリア信号を発生するキャリアマルチプレクサとを含む、請求項1に記載のディレイロックドループ。

【請求項3】 前記遅延回路は、

ファイン遅延素子と、

前記ファイン遅延回路よりも単位あたりの遅延量が多いコース遅延素子とを含む、

前記ファイン遅延素子は、前記バイナリコード変換器に格納される前記バイナリコードにより遅延量が調整される、

前記コース遅延素子は、前記グレイコードレジスタに格納される前記グレイコードにより遅延量が調整される、請求項2に記載のディレイロックドループ。

【請求項4】 外部クロックを受けて第1の内部クロックを出力する入力バッファと、

前記第1の内部クロックを遅延して第2の内部クロックを出力する遅延回路と、

前記第1の内部クロックと前記第2の内部クロックとの位相差を検出する検出器と、

前記検出器の出力に応じて前記遅延回路の遅延量を調整する信号を発生する、グレイコードを用いたグレイコードカウンタとを含むディレイロックドループを備える、半導体装置。

【請求項5】 前記グレイコードカウンタは、

前記グレイコードを格納するグレイコードレジスタと、

前記グレイコードをバイナリコードに変換するバイナリコード変換器と、

前記バイナリコード変換器に格納されるバイナリコードによりアップキャリア信号とダウンキャリア信号とを発生するアップキャリア／ダウンキャリア発生器と、

前記検出器の結果に応じて、前記アップキャリア信号および前記ダウンキャリア信号から、前記グレイコードレジスタにおけるグレイコードを更新するためのキャリア

信号を発生するキャリアマルチプレクサとを含む、請求項4に記載の半導体装置。

【請求項6】 前記遅延回路は、

ファイン遅延素子と、

前記ファイン遅延回路よりも単位あたりの遅延量が多いコース遅延素子とを含む、

前記ファイン遅延素子は、前記バイナリコード変換器に格納される前記バイナリコードにより遅延量が調整される、

前記コース遅延素子は、前記グレイコードレジスタに格納される前記グレイコードにより遅延量が調整される、請求項5に記載の半導体装置。

【請求項7】 前記第2の内部クロックに応じて動作する、信号を外部から受ける入力回路をさらに備える、請求項5に記載の半導体装置。

【請求項8】 複数のメモリセルを含むメモリセルアレイをさらに備え、

前記入力回路は、前記メモリセルアレイにおけるデータ書込／読出のための信号を受ける、請求項7に記載の半導体装置。

【請求項9】 前記第2の内部クロックに応じて動作する、信号を外部に出力するための出力回路をさらに備える、請求項5に記載の半導体装置。

【請求項10】 複数のメモリセルを含むメモリセルアレイをさらに備え、

前記出力回路は、前記メモリセルアレイから読出したデータを外部に出力する、請求項9に記載の半導体装置。

【請求項11】 第1の外部クロックと前記第1の外部クロックと位相が相補になっている第2の外部クロックとが少なくとも入力され、前記第1の外部クロックの電位と前記第1の外部クロックの電位とが等しくなる前記第1の外部クロックの立上がりエッジのタイミングにおいて第1の内部クロックを出力する第1の入力バッファと、

前記第1の外部クロックと前記第2の外部クロックとが少なくとも入力され、前記第1の外部クロックの電位と前記第1の外部クロックの電位とが等しくなる前記第2の外部クロックの立上がりエッジのタイミングにおいて第2の内部クロックを出力する第2の入力バッファと、前記第1の内部クロックを遅延して第3の内部クロックを出力する第1の遅延回路と、

前記第2の内部クロックを遅延して第4の内部クロックを出力する第2の遅延回路と、

前記第1の内部クロックと前記第3の内部クロックとの位相差を検出する検出器と、

前記検出器の出力に応じて前記第1の遅延回路の遅延量と前記第2の遅延回路の遅延量とを調整する信号を発生するグレイコードを用いたグレイコードカウンタとを含むディレイロックドループを備える、半導体装置。

【請求項12】 前記第3の内部クロックおよび前記第

4の内部クロックに応じて動作する、信号を外部に出力するための出力回路をさらに備える、請求項11に記載の半導体装置。

【請求項13】 複数のメモリセルを含むメモリセルアレイをさらに備え、

前記出力回路は、前記メモリセルアレイから読出したデータを第3および第4の内部クロックに応じて外部に出力する、請求項12に記載の半導体装置。

【請求項14】 クロック同期により動作するシステムのための制御方法であって、

外部クロックを受けて第1の内部クロックを発生する入力バッファステップと、

前記第1の内部クロックを遅延して第2の内部クロックを出力する遅延ステップと、

前記第1の内部クロックと前記第2の内部クロックとの位相差を検出する検出ステップと、

グレイコードを用いて、前記検出ステップにおける検出結果に応じて前記遅延ステップにおける遅延量を決定するステップとを備える、クロック同期により動作するシステムのための制御方法。

【請求項15】 前記遅延量を決定するステップは、前記グレイコードをバイナリコードに変換するバイナリコード変換ステップと、

前記バイナリコードによりアップキャリー信号とダウンキャリー信号とを発生するアップキャリー／ダウンキャリー発生ステップと、

前記検出ステップの結果に応じて、アップキャリー信号および前記ダウンキャリー信号から前記グレイコードを更新するためのキャリー信号を発生するステップとを含む、請求項14に記載のクロック同期により動作するシステムのための制御方法。

【請求項16】 前記第2の内部クロックに応じてデータを外部に出力するステップをさらに備える、請求項15に記載のクロック同期により動作するシステムのための制御方法。

【請求項17】 前記第2の内部クロックに応じて外部からデータを受けるステップをさらに備える、請求項15に記載のクロック同期により動作するシステムのための制御方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、クロック同期により動作するシステムにおいて同期を正確にとるために用いられるディレイロックドループ、当該ディレイロックドループを含む半導体装置およびクロック同期により動作するシステムのための制御方法に関するものである。

【0002】

【従来の技術】 クロック同期によって動作するシステムにおいて、ある部品がクロックに同期した読出命令を受取ってから、当該読出命令に回答して読出データをクロ

ック同期してシステムバスに返す場合を考える。なお、以下において記号ADRは、バイナリコードを示す。また、記号 $ADR < k >$ は、バイナリコードにおけるkビット目を意味し、レジスタ値 $ADR < k >$ またはバイナリコード $ADR < k >$ と称す。

【0003】 システムクロック周期を“T”とし、部品が読出命令を受取ってから読出データを準備するためにT0だけ時間を要するとする。クロック周期Tと期間T0との大小関係は問わないものとし、ある整数Nに対し、 $N \times T \leq T0 < (N+1) \times T$ という関係が成り立っているものとする。この場合の出力の方法としては、以下の方法が考えられている。

【0004】 図17を参照して、時刻 $t_1$ （クロックCLKextの立上がりエッジ0）においてクロックCLKextに同期して読出命令が発せられ、読出データが準備され、次のクロックCLKextの立上がりエッジ1（時刻 $t_2 : t_2 = t_1 + T$ ）に同期してデータが出力されている。

【0005】 部品がクロックCLKextを受取ってから、準備されたデータを出力バッファに出力するまでにはTdだけ時間を要する。このため、クロックエッジ1よりも実際の出力データはTdだけ遅れて出力される。

【0006】 期間Tdの内訳としては、図18に示すように、部品内部において出力バッファを駆動するクロックCLKinをクロックCLKextから発生するための時間Tinと、クロックCLKinによって出力バッファが動作し始めてから出力負荷であるシステムバスを所定の論理閾値を超えるまで駆動するための出力バッファ駆動時間Toutとが含まれる（ $Td = Tin + Tout$ ）。

【0007】 したがって、当該システムにおいては、実際にはクロックエッジ1よりもTdだけ遅れてからでないとデータはシステムバスに出力されないため、次のクロックエッジ2においてデータを取込むようにシステムを設計することになる。

【0008】 このような動作である場合、システムクロック周期TがTdに比べて等しい、またはTがTdよりも小さいくらいまで高周波になってくると、図19に示すような問題が生じる。図19においては、クロックエッジ1においてデータが出力し始めたにもかかわらず、遅延時間TdがTよりも大きいため、クロックエッジ2においてまだデータがバス上に伝達されていない。したがって、次のクロックエッジ3まで待たないとシステムはデータを取込むことができない。言換えると、クロックエッジ2においてデータを取込むような設計のままだと、システムが誤動作を起こすことになる。

【0009】 特に、近年開発されているDDR・SDRAM（ダブルデータレート・シンクロナスタティックランダムアクセスメモリ）においては、クロックの立上がりエッジと立下がりエッジとの両方でデータを出力す

る。このため、図20に示すように、上記した出力手法をDDR・SDRAMに適用したならば、クロック周期の $1/2$ が $T_d$ とほぼ等しくなる時点で上記問題と同じ問題が顕在化することになる。

【0010】このような問題点を解決するために従来用いられているのが、ディレイ・ロックド・ループ（以下、DLLと記す）である。DLLを用いた部品においては、図21に示すように、部品がシステムクロック $CLK_{ext}$ を受取って、当該システムクロック $CLK_{ext}$ を部品内部で遅延素子により $T_{d111}$ だけ遅延させる。そして、システムクロックの立上がりエッジよりも出力バッファ駆動時間相当の時間（ $T_{out}' = T_{out}$ ）だけ手前で立上がるような内部クロック $CLK_{in2}$ を発生する。

【0011】内部クロック $CLK_{in2}$ を用いて出力バッファを駆動することにより、システムクロック $CLK_{ext}$ のエッジで読出データがバス上に出力されているという状態が実現できる。すなわち、 $T_{d111} + T_{out}' = m \times T$ （ $m$ は、1以上の整数）が成立する。

【0012】このように、DLLは出力タイミングをクロックに同期させることができる。同様に、部品がシステムバスから各種制御信号や入力データを取込むための入力バッファに対してもDLLを用いることができる。

【0013】DLLを用いない場合の入力バッファの動作タイミングを、図22を用いて説明する。図22に示されるように、DLLを用いない入力バッファにおいては、クロック $CLK_{ext}$ を受けてから部品内部で内部クロック $CLK_{in}$ を発生するのに $T_{in}$ だけの時間を要している。

【0014】このため、外部から入力される制御信号 $SIG$ は、 $T_{in}$ 相当の時間 $T_{in}$ だけ内部で遅れた信号 $SIG_{in}$ になり、信号 $SIG_{in}$ は内部クロック $CLK_{in}$ のエッジタイミングでラッチ回路910によりラッチされ確定することになる。図において、確定した制御信号を示している。

【0015】ラッチ回路910は、図23に示すように、インバータ $IV1 \sim IV4$ およびNAND回路 $N1 \sim N4$ を含む。インバータ $IV1, IV3$ は、内部クロック $CLK_{in}$ および当該内部クロック $CLK_{in}$ を反転したクロック $\neg CLK_{in}$ に応じて動作する。

【0016】したがって、DLLを用いない場合、システムクロック $CLK_{ext}$ のエッジよりも $T_{in}$ 以上遅れてからでないと部品は制御信号を使用することができず、部品自体が高速動作に適さなくなる。

【0017】このような遅延を補償するため、上記したDLLを用いることになる。DLLを用いた動作波形について、図24を用いて説明する。図24を参照して、システムクロック $CLK_{ext}$ を部品内部で遅延素子により $T_{d112}$ だけ遅延させ、システムクロック $CLK_{ext}$ のエッジと同じ位置にエッジを持つ内部クロック

$CLK_{in3}$ を生成する。そして、当該内部クロック $CLK_{in3}$ を制御信号 $SIG$ をラッチするために用いたとすれば、上記した場合よりも、 $T_{in}$ 相当分だけの時間だけ早く制御信号 $SIG$ をラッチすることができる。これにより高速動作に適したシステム設計となる。この場合、上記した説明により $T_{d112} = m \times T$ が成立する。

【0018】ここで、従来から用いられてきたDLLを構成する回路の構成例を、図25を用いて説明する。従来のDLL9000は、図25に示すように、クロックバッファ1、ファイン遅延素子3、コース遅延素子5、デコーダ70、遅延レジスタであるバイナリカウンタ80、位相比較器9、タイミングクロック発生器10、パルス発生器11およびレプリカ回路13を含む。

【0019】外部クロック $CLK_{ext}$ がクロックバッファ1に入力される。クロックバッファ1は、クロック $BUFFCLK$ を出力する。外部クロック $CLK_{ext}$ からクロック $BUFFCLK$ までの時間は $T_{in}$ である。

【0020】クロック $BUFFCLK$ は、ファイン遅延素子3に入力される。ファイン遅延素子3は、クロック $BUFFCLK$ を $T_{fine}$ だけ遅らせて、クロック $CLK_A$ を出力する。クロック $BUFFCLK$ からクロック $CLK_A$ までの遅延時間 $T_{fine}$ は可変であって、ファイン遅延素子3に入力される3ビットのレジスタ値 $ADR < 0 : 2 >$ の値によって微小な刻み $T_f$ （約40 psec刻み）で変化する。

【0021】クロック $CLK_A$ はコース遅延素子5に入力される。コース遅延素子5は、クロック $CLK_A$ を $T_{coarse}$ だけ遅らせて、クロック $CLK_B$ を出力する。クロック $CLK_A$ からクロック $CLK_B$ までの遅延時間 $T_{coarse}$ は可変であって、6ビットのレジスタ値 $ADR < 3 : 8 >$ によって粗い刻み $T_c$ で変化する。なお、 $T_c$ は、 $T_f$ の8倍の大きさに設定しておく。

【0022】ファイン遅延素子3およびコース遅延素子5の遅延量を決定するレジスタ値は、バイナリカウンタ80から出力される。

【0023】クロックがファイン遅延素子3とコース遅延素子5とを通過するのに、回路構成上、少なくとも遅延時間 $T_x$ かかる。ファイン遅延素子3とコース遅延素子5とを通過する以上、たとえばレジスタ値が最短遅延時間の値であったとしても $T_x$ だけ遅延してしまう。

【0024】さて、従来のDLLは、ファイン遅延素子3およびコース遅延素子5のいずれにおいても、バイナリコードにより遅延時間を設定していた。一例としては、ファイン遅延素子3における遅延時間は、レジスタ値 $ADR < 0 : 2 >$ が“001”だと $T_f$ 、“010”だと $2T_f$ 、“011”だと $3 \times T_f$ になる。また、コース遅延素子5における遅延時間は、レジスタ値 $ADR$

<3:8>が“000111”だと $7 \times T_c$ 、“001000”だと $8 \times T_c$ になる。

【0025】したがって、上記した回路構成上発生する遅延時間 $T_x$ も考慮に入れると、 $T_{coarse}$ を決定するレジスタ値が“001001”であり $T_{fine}$ を決定するレジスタ値が“011”の場合、クロックBUFFCLKからクロックCLKBまでのトータルの遅延時間は、 $T_{fine} + T_{coarse} = T_x + 3 \times T_f + 9 \times T_c$ になる。

【0026】以降の説明において、コース遅延素子5に入力されるレジスタ値が“001001”であり、ファイン遅延素子3に入力されるレジスタ値が“011”の場合には、レジスタ値“001001011”と表記する。

【0027】クロックCLKBは、出力バッファ駆動用の内部クロックCLKin2と入力バッファ駆動用の内部クロックCLKin3とを発生するパルス発生器11およびレプリカ回路13に入力されている。内部クロックCLKin2は、図21で説明した役割を果たす。これにより、外部クロックCLKextの立上がりエッジで出力バッファの出力がシステムバスに伝達されることになる。

【0028】レプリカ回路13は、入力バッファ遅延時間 $T_{in}$ と出力バッファ駆動時間 $T_{out}$ との和を模擬した固定遅延時間( $T_{in}' + T_{out}'$ )だけクロックCLKBを遅らせて、クロックFBCLKを発生する。

【0029】したがって、クロックBUFFCLKからクロックFBCLKまでの遅延時間は、 $T_{fine} + T_{coarse} + T_{in}' + T_{out}'$ となる。上記した外部クロックCLKextからクロックFBCLKの発生に至るまでの各種信号の関係は、図26に示すとおりである。

【0030】ここで、DLLが、所望の遅延時間を満たすように動作している場合の波形について図27を用いて説明する。図27に示す例では、クロックBUFFCLKの立上がりエッジとクロックFBCLKとの立上がりエッジが同時になっている。クロックBUFFCLKよりも $T_{in}$ だけ前に外部クロックCLKextがあり、クロックFBCLKよりも( $T_{in}' + T_{out}'$ )だけ前に内部クロックCLKin2の立上がりエッジがある。このため、外部クロックCLKextの立上がりエッジよりも $T_{out}'$ だけ前に内部クロックCLKin2の立上がりエッジがあることになる。これは、上述した図21において説明したタイミングに相当している。ここで、 $T_{in} = T_{in}'$ 、 $T_{out} = T_{out}'$ となるようにレプリカ回路13における遅延時間は設計されている。

【0031】ここで、DLLの最適遅延時間を決定するためにはクロックBUFFCLKとクロックFBCLK

との立上がりエッジの先後を検出する回路が必要になる。これが、図25に示される位相比較器9である。位相比較器9は、クロックBUFFCLKとクロックFBCLKとの位相差を検出して、信号UPF、DNFを出力する。

【0032】ここで、信号UPF、DNFについて、図28(a)、図28(b)を用いて説明する。クロックFBCLKの位相がクロックBUFFCLKの位相よりも進んでいる場合、図28(a)に示すように、位相比較器9の出力は「アップ」(UPF=“H”、DNF=“L”)になる。また、クロックFBCLKの位相がクロックBUFFCLKの位相よりも遅れている場合には、図28(b)に示すように、位相比較器9の出力は「ダウン」(DNF=“H”、UPF=“L”)になる。

【0033】位相比較器9の構成の一例を図30に示す。位相比較器9は、図30に示すように、NAND回路N5~N10を含む。NAND回路N5に入力されるクロックBUFFCLKとNAND回路N6に入力されるクロックFBCLKとにより、NAND回路N9から信号UPFが、NAND回路N10から信号DNFがそれぞれ出力される。

【0034】図25を参照して、タイミングクロック発生器10は、信号UPF、DNFを受けて、カウンタ更新クロックCLKCNTとアップ信号UPとダウン信号DNとを発生する。

【0035】バイナリカウンタ80は、カウンタ更新クロックCLKCNTに応じて、アップ信号UP、ダウン信号DNに基づき、レジスタ値 $ADR < 0:2 >$ 、 $ADR < 3:8 >$ を出力する。

【0036】位相比較器9の出力が「アップ」であると、レジスタ値が増加する。たとえば、“000000010”が、“000000011”になる。また、位相比較器9の出力が「ダウン」であると、レジスタ値が減少する。たとえば、“000000011”が、“000000010”になる。

【0037】デコーダ70は、レジスタ値 $ADR < 3:8 >$ をデコードする。デコーダ70は、図29に示すように、AND回路90#0~90#7および91#0~91#15を含む。

【0038】AND回路90#0は、バイナリコード $ADR < 0 >$ を反転したバイナリコード/ $ADR < 0 >$ と、バイナリコード $ADR < 1 >$ を反転したバイナリコード/ $ADR < 1 >$ とを受け、AND回路90#1は、バイナリコード $ADR < 0 >$ とバイナリコード/ $ADR < 1 >$ とを受ける。

【0039】AND回路90#2は、バイナリコード/ $ADR < 0 >$ と、バイナリコード $ADR < 1 >$ とを受け、AND回路90#3は、バイナリコード $ADR < 0 >$ とバイナリコード $ADR < 1 >$ とを受ける。

【0040】AND回路90#4は、バイナリコードADR<2>を反転したバイナリコード／ADR<2>と、バイナリコードADR<3>を反転したバイナリコード／ADR<3>とを受け、AND回路90#5は、バイナリコードADR<2>とバイナリコード／ADR<3>とを受ける。

【0041】AND回路90#6は、バイナリコード／ADR<2>と、バイナリコードADR<3>とを受け、AND回路90#7は、バイナリコードADR<2>とバイナリコードADR<3>とを受ける。

【0042】AND回路91#i (i=0~3)は、AND回路90#iの出力X<i>とAND回路90#4の出力X<4>とを入力に受ける。AND回路91#i+4 (i=0~3)は、AND回路90#iの出力X<i>とAND回路90#5の出力X<5>とを受ける。AND回路91#i+8 (i=0~3)は、AND回路90#iの出力X<i>とAND回路90#6の出力X<6>とを入力に受ける。AND回路91#i+12 (i=0~3)は、AND回路90#iの出力X<i>とAND回路90#7の出力X<7>とを受ける。AND回路91#k (k=0~15)の出力に応じて、コース遅延素子5の遅延段数(たとえば、k個)が決定される。ここでは4ビットバイナリデコードについて例示したが、ADR<3>~ADR<8>を受ける6ビットバイナリデコードも同様の回路により実現されることは当業者によって容易に理解されるところである。

【0043】外部クロックを遅延しすぎると、出力「ダウン」により遅延時間が短くなり、遅延が足りなくなると出力「アップ」により遅延時間が長くなる。この結果、DLL9000は、所望の遅延時間で安定することになる。

【0044】DLLは、このように、クロックBUFFCLKとクロックFBCLKとの立上がりエッジの先後を検出しつつ、クロックBUFFCLKとクロックFBCLKとの立上がりエッジが同時になる状態を保持するよう遅延素子3および5の遅延時間を適宜調節する。

【0045】

【発明が解決しようとする課題】ところで、図28 (a) および図28 (b) のように位相差が明確である場合には何の問題も生じない。しかしながら、DLL9000は上記したようにクロックBUFFCLKとクロックFBCCLKとの位相差をモニタし、位相差がほとんどない状態を保持するように遅延素子3および5を調整すべく動作せねばならない。

【0046】したがって、必然的にDLL9000が動作している通常状態では、クロックBUFFCLKとクロックFBCCLKとの位相差はほとんどない。このような場合には、位相比較器9は、「アップ」または「ダウン」のいずれかを出力すべきなのかを決定するのに非常に長い時間を要する状態がある確率で発生する。

【0047】これは、位相比較器9を含めフリップフロップ一般について言えることで、メタステーブル状態として知られている現象である。

【0048】しかしながら、「CMOS超LSIの設計」(培風館、1989年刊、p. 128)にあるように、メタステーブル状態の発生確率を完全に0にすることは現実的に不可能である。このような場合に、従来のようなバイナリカウンタ80を用いると誤動作を起こす可能性がある。

【0049】たとえば、現在のレジスタ値が“01111111”である場合、次のレジスタ値は、「アップ」であれば“1000000000”になるが、「ダウン」であれば、“0111111110”になる。

【0050】つまり、この例ではMSBビットからLSBビットの1つ手前までの合計8ビットの状態が、「アップ」か「ダウン」かによって異なることになる。言換えると、「アップ」なら、9ビットのキャリーが“1”なのに、「ダウン」であるならば、LSBビット以外のキャリーは“0”であり、「アップ」か「ダウン」かでキャリーの値が大きく異なる。

【0051】こうした瞬間にメタステーブル状態が発生すると、位相比較器9は、「アップ」か「ダウン」か、すなわちレジスタのキャリーは“0”なのか“1”なのか不定になる。この結果として、次のレジスタ状態が不定になる。実際には、次のレジスタ状態はランダムな値になる。

【0052】すなわち、DLLの遅延時間がある値から突然大きく異なる別の値になってしまうという結果をもたらす。たとえば、レジスタ値が“01111111”から“010101010”になった場合には、遅延素子の遅延時間 $T_{coarse} + T_{fine}$ は $31 \times T_c + 7 \times T_f$ から $21 \times T_c + 2 \times T_f$ になる。

【0053】したがって、所望の遅延状態で安定していたはずのDLLにおいて、遅延時間が、突然大きくずれてしまい、系全体が誤動作状態を起こしてしまう。

【0054】そこで、本発明はかかる問題を解決するためになされたものでありその目的は、不連続な遅延の飛びを最小化することができるディレイロックドループおよび当該ディレイロックドループを備える半導体装置を提供する。

【0055】

【課題を解決するための手段】この発明のある局面によるディレイロックループは、第1クロックを遅延して第2クロックを出力する遅延回路と、前記第1クロックと前記第2クロックとの位相差を検出する検出器と、前記検出器の出力に応じて前記遅延回路の遅延量を調整する信号を発生する、グレイコードを用いたグレイコードカウンタとを備える。

【0056】好ましくは、グレイコードカウンタは、グレイコードを格納するグレイコードレジスタと、グレイ

コードをバイナリコードに変換するバイナリコード変換器と、バイナリコード変換器に格納されるバイナリコードによりアップキャリア信号とダウンキャリア信号とを発生するアップキャリア／ダウンキャリア発生器と、検出器の結果に応じて、アップキャリア信号およびダウンキャリア信号から、グレイコードレジスタにおけるグレイコードを更新するためのキャリア信号を発生するキャリアマルチプレクサとを含む。

【0057】特に、遅延回路は、ファイン遅延素子と、ファイン遅延回路よりも単位あたりの遅延量が大いコース遅延素子とを含み、ファイン遅延素子は、バイナリコード変換器に格納されるバイナリコードにより遅延量が調整され、コース遅延素子は、前記グレイコードレジスタに格納されるグレイコードにより遅延量が調整される。

【0058】この発明のある局面による半導体装置は、外部クロックを受けて第1の内部クロックを出力する入力バッファと、第1の内部クロックを遅延して第2の内部クロックを出力する遅延回路と、第1の内部クロックと第2の内部クロックとの位相差を検出する検出器と、検出器の出力に応じて遅延回路の遅延量を調整する信号を発生する、グレイコードを用いたグレイコードカウンタとを含むディレイロックドループを備える。

【0059】好ましくは、グレイコードカウンタは、グレイコードを格納するグレイコードレジスタと、グレイコードをバイナリコードに変換するバイナリコード変換器と、バイナリコード変換器に格納されるバイナリコードによりアップキャリア信号とダウンキャリア信号とを発生するアップキャリア／ダウンキャリア発生器と、検出器の結果に応じて、アップキャリア信号およびダウンキャリア信号から、グレイコードレジスタにおけるグレイコードを更新するためのキャリア信号を発生するキャリアマルチプレクサとを含む。

【0060】特に、遅延回路は、ファイン遅延素子と、ファイン遅延回路よりも単位あたりの遅延量が大いコース遅延素子とを含み、ファイン遅延素子は、バイナリコード変換器に格納されるバイナリコードにより遅延量が調整され、コース遅延素子は、前記グレイコードレジスタに格納されるグレイコードにより遅延量が調整される。

【0061】好ましくは、半導体装置は、第2の内部クロックに応じて動作する、信号を外部から受ける入力回路または、第2の内部クロックに応じて動作する、信号を外部に出力するための出力回路をさらに備える。特に、半導体装置は、複数のメモリセルを含むメモリセルアレイをさらに備え、入力回路は、前記メモリセルアレイにおけるデータ書込／読出のための信号を受け、出力回路は、前記メモリセルアレイから読出したデータを外部に出力する。

【0062】この発明のさらなる局面による半導体装置

は、第1の外部クロックと第1の外部クロックと位相が相補になっている第2の外部クロックとが少なくとも入力され、第1の外部クロックの電位と第1の外部クロックの電位とが等しくなる第1の外部クロックの立上がりエッジのタイミングにおいて第1の内部クロックを出力する第1の入力バッファと、第1の外部クロックと第2の外部クロックとが少なくとも入力され、第1の外部クロックの電位と第1の外部クロックの電位とが等しくなる第2の外部クロックの立上がりエッジのタイミングにおいて第2の内部クロックを出力する第2の入力バッファと、第1の内部クロックを遅延して第3の内部クロックを出力する第1の遅延回路と、第2の内部クロックを遅延して第4の内部クロックを出力する第2の遅延回路と、第1の内部クロックと第3の内部クロックとの位相差を検出する検出器と、検出器の出力に応じて第1の遅延回路の遅延量と第2の遅延回路の遅延量とを調整する信号を発生するグレイコードを用いたグレイコードカウンタとを含むディレイロックドループを備える。

【0063】好ましくは、半導体装置は、第3の内部クロックおよび第4の内部クロックに応じて動作する、信号を外部に出力するための出力回路をさらに備える。

【0064】特に、半導体装置は、複数のメモリセルを含むメモリセルアレイをさらに備え、出力回路は、メモリセルアレイから読出したデータを第3および第4の内部クロックに応じて外部に出力する。

【0065】この発明のさらなる局面によるクロック同期により動作するシステムにおける制御方法は、外部クロックを受けて第1の内部クロックを発生する入力バッファステップと、第1の内部クロックを遅延して第2の内部クロックを出力する遅延ステップと、第1の内部クロックと第2の内部クロックとの位相差を検出する検出ステップと、グレイコードを用いて、検出ステップにおける検出結果に応じて遅延ステップにおける遅延量を決定するステップとを備える。

【0066】好ましくは、遅延量を決定するステップは、グレイコードをバイナリコードに変換するバイナリコード変換ステップと、バイナリコードによりアップキャリア信号とダウンキャリア信号とを発生するアップキャリア／ダウンキャリア発生ステップと、検出ステップの結果に応じて、アップキャリア信号および前記ダウンキャリア信号から前記グレイコードを更新するためのキャリア信号を発生するステップとを含む。

【0067】特に、制御方法は、第2の内部クロックに応じてデータを外部に出力するステップをさらに備える。または、制御方法は、第2の内部クロックに応じて外部からデータを受けるステップをさらに備える。

【0068】

【発明の実施の形態】以下、本発明の実施の形態について、図を用いて説明する。図中同一または相当部分には同一記号または符号を付しその説明を省略する。



【0069】[第1の実施の形態]第1の実施の形態におけるディレイロックドロープは、不連続な遅延の飛びを最小化することができる構成を有する。より具体的には、DLLにおいて、遅延レジスタとしてグレイコードカウンタ(グレイコード:交番符号)を用いる。

【0070】これにより、キャリーを常に1ビットしかたてないようにすることで、仮にメタステーブル状態が起きても遅延時間の飛び(不連続な飛び)を最小化することができる。以下において、記号GADRは、グレイコードを意味する。また、記号 $GADR<k>$ は、グレイコードにおけるkビット目を意味し、レジスタ値 $ADR<k>$ またはグレイコード $GADR<k>$ と称す。

【0071】ここで、4ビットのグレイコードについて、図1を用いて説明する。図1では、デシマル(10進コード)に対するバイナリコード(2進コード)とグレイコードとをそれぞれ示している。図中、“↑”(上向き矢印)は、「ダウン」方向を、“↓”(下向き矢印)は、「アップ」方向を表わしている。また、記号ADRは、バイナリコードを、記号GADRは、グレイコードを意味している。記号“.”(黒丸)は、バイナリコードおよびグレイコードのそれぞれについてキャリーがどのビットに立っているかを示している。

【0072】グレイコードの作成方法は、以下に示すとおりである。まず、デシマル“0”をグレイ“0000”に割当てたとする。この点から、アップ方向に各ビットがどのように“0”または“1”にアサインされていくか説明する。

【0073】グレイコードの0ビット目(LSB)は、アップ方向に0110を繰返す。1ビット目は、アップ方向に00111100を繰返す。2ビット目は、アップ方向に00001111110000を繰返す。以下同様に、nビット目は、アップ方向に“0”を2のn乗回、“1”を2の(n+1)乗回、“0”を2のn乗回繰返す。

【0074】なお、図1に示すグレイコードにおいて、nビット目の論理値を逆転させたものもグレイコードになる。

【0075】バイナリコード、グレイコードのどちらについても、M番目のコードのアップキャリーがiビット目に立っている場合、M+1番目のコードのダウンキャリーもiビット目に立つ。したがって、図1において、M番目のコードとM+1番目のコードとの間にある記号“.”は、M番目のコードから見るとアップキャリーに相当し、M+1番目のコードから見るとダウンキャリーになる。

【0076】バイナリコードでは、アップまたはダウン方向のいずれにおいても複数ビットにキャリーが立っているが、グレイコードでは、アップおよびダウン方向のいずれにおいても1つのビットにのみキャリーが立つことがわかる。

【0077】たとえば、バイナリコード“0110”(デシマルコード“6”)を考える。バイナリコード“0110”のダウンキャリーは、0ビット目と1ビット目に立っており、アップキャリーは、0ビット目に立っている。一方、1つ前のデシマルコード“5”(バイナリコード“0101”)のアップキャリーは、0ビット目と1ビット目とに立っており、1つ後のデシマルコード“7”(バイナリコード“0111”)のダウンキャリーは、0ビットに立っている。

【0078】これに対し、グレイコード“0101”(デシマルコード“6”)について考察する。グレイコード“0101”のダウンキャリーは、1ビット目に立っており、アップキャリーは、0ビット目に立っている。1つ前のデシマルコード“5”(グレイコード“0111”)のアップキャリーは、1ビット目に立っており、1つ後のデシマルコード“7”(グレイコード“0100”)のダウンキャリーは、0ビット目に立っている。すなわち、グレイコードでは、キャリーが常に1ビットしか立たないことがわかる。

【0079】4ビットのグレイコードを4ビットのバイナリコードに変換する変換式の一例を、図2を用いて説明する。図2において、Exorとは、排他的論理和を意味する。Exor回路は、入力される複数の信号のうち“1”の数が奇数であれば“1”を、偶数であれば“0”を出力する。

【0080】たとえば、 $Exor(GADR<3>, GADR<2>)$ は、バイナリコード $ADR<2>$ とし、 $Exor(GADR<3> \sim GADR<1>)$ は、バイナリコード $ADR<3>$ とする。すなわち、図2に示される変換式で、グレイコード $GADR<3>:0>$ をバイナリコード $ADR<3>:0>$ に変換することが可能になる。

【0081】同様に、nビットのグレイコードをnビットのバイナリコードに変換する変換式を、図3を用いて説明する。たとえば、グレイコード $GADR<n-1>$ は、バイナリコード $ADR<n-1>$ とし、 $Exor(GADR<n-1> \sim GADR<n-3>)$ は、バイナリコード $ADR<n-3>$ とする。図3に示される変換式で、nビットのグレイコードをnビットのバイナリコードに変換することが可能になる。

【0082】次に、グレイコードのキャリーがどのビットに立つかについて、図4を用いて説明する。図4では、グレイコード“0101”をデシマル“6”(バイナリコード“0110”)に対応させた例を示している。図中、キャリービットとバイナリコードにおけるビットとの関係を、矢印で表わしている。

【0083】グレイコード“0101”のダウンキャリーは、対応するバイナリコード“0110”のうち、最もLSB側の“1”が立っているビット、つまり1ビット目に立つ。また、アップキャリーは、対応するバイナ

リコード“0110”のうち、最もLSB側の“0”が立っているビット、つまり0ビット目に立つ。

【0084】一般化すると、あるグレイコードについて考えたとき、当該グレイコードのダウンキャリアは、対応するバイナリコードのうち最もLSB側の“1”が立っているビットに立つ。また、アップキャリアは、対応するバイナリコードのうち、最もLSB側の“0”が立っているビットに立つ。これにより、グレイコードにおいては、アップキャリア、ダウンキャリアのそれぞれは、1ビットにしか立たないことがわかる。

【0085】ここで、第1の実施の形態によるグレイコードカウンタ(9ビット)の構成の一例を、図5を用いて説明する。グレイコードカウンタ81は、図5に示すように、グレイコードレジスタ810、バイナリコード発生器811、アップ・ダウンキャリア発生器812およびキャリアマルチプレクサ813を含む。

【0086】グレイコードレジスタ810は、9ビットのグレイコードGADR<0>～GADR<8>を格納する。バイナリコード発生器811は、Exor回路E0～E7を含む。

【0087】バイナリコード発生器811は、グレイコードレジスタ810から9ビットのグレイコードGADR<0>～GADR<8>を受けて、図3で説明した変換式にしたがってバイナリコードADR<0>～ADR<8>を生成する。

【0088】アップ・ダウンキャリア発生器812は、バイナリコードADR<0>～ADR<8>を受けて、図4で説明した変換式に従い、アップキャリアUC<0>～UC<8>またはダウンキャリアDC<0>～DC<8>を生成する。

【0089】上述したように、アップキャリアUC<0>～UC<8>のうちの1ビットのみが“1”であり、他は“0”である。または、ダウンキャリアDC<0>～DC<8>のうちの1ビットのみが“1”であり、他は“0”である。

【0090】キャリアマルチプレクサ813は、信号UPおよびDNを受ける複数のマルチプレクサ(記号MUX)を含む。信号UPとDNとは、相対的に反対の論理レベルを有する。

【0091】信号UP=“1”、信号DN=“0”であれば、次はカウンタをアップさせねばならないので、アップキャリアUC<0>～UC<8>をキャリアC<0>～C<8>としてスルーする。一方、信号UP=“0”、信号DN=“1”ならば、次は、カウンタをダウンさせねばならないので、ダウンキャリアDC<0>～DC<8>をキャリアC<0>～C<8>としてスルーする。

【0092】キャリアC<0>～C<8>のうち1ビットのみが“1”であり、他は“0”である。上記したキャリアC<0>～C<8>の発生過程は、図6に示すと

おりである。このようにして生成されたキャリアC<0>～C<8>が、グレイコードレジスタ810に入力される。そして、次のカウンタ更新クロックCLKCNTのタイミングで、入力されるキャリアC<0>～C<8>に基づき、グレイコードGADR<0>～GADR<8>のうちの1ビットが新しい値に更新される。

【0093】ここで、第1の実施の形態によるグレイコードレジスタ810の回路構成の一例を、図7を用いて説明する。図7は、グレイコードGADR<n>に対応するレジスタの構成を示している(n=0～8)。グレイコードGADR<n>対応のレジスタは、NAND回路N20、インバータIV20～IV25、ならびにゲートG20およびG21を含む。

【0094】NAND回路N20は、カウンタ更新クロックCLKCNTとキャリアC<n>とを受け、信号/Cを出力する。インバータIV20は、信号/Cを反転し信号Cを出力する。

【0095】インバータIV21およびIV22はラッチ回路を構成する。同様に、インバータIV23およびIV24は、ラッチ回路を構成する。ゲートG20およびG21は、信号Cおよび/Cに応じて入力信号を出力信号にスルーし、または入力信号と出力信号とを切断する。

【0096】インバータIV25は、グレイコードGADR<n>を出力する出力ノードOUTとゲートG20との間に接続される。インバータIV21およびIV22を含むラッチ回路は、ゲートG20とG21との間に接続される。インバータIV23およびIV24を含むラッチ回路は、ゲートG21とノードOUTとの間に接続される。

【0097】次に、第1の実施の形態によるバイナリコード発生器811の回路構成の一例を、図8を用いて説明する。バイナリコード発生器811は、図8に示すように、EXOR回路E9a、E9b、E9c、…を含む。Exor回路は、図3で説明したグレイコード/バイナリコード変換式に対応するように配置される。

【0098】たとえば、グレイコードGADR<8>は、バイナリコードADR<8>として出力される。グレイコードGADR<8>とGADR<7>とを受けるEXOR回路E9bの出力がバイナリコードADR<7>として出力される。グレイコードGADR<6>～GADR<8>を入力に受けるEXOR回路E9cの出力が、バイナリコードADR<6>として出力される。

【0099】次に、第1の実施の形態によるアップ・ダウンキャリア発生器812の回路構成の一例を、図33を用いて説明する。アップ・ダウンキャリア発生器812は、図33に示すように、複数の論理回路40、41、42、…を含む。アップ・ダウンキャリア発生器812における複数の論理回路は、図4で説明したキャリアビットの位置を求める方式に従い配置される。アップ

・ダウンキャリー発生器812により、ダウンキャリーDC<0>、DC<1>、…、アップキャリーUC<0>、UC<1>、…が出力される。

【0100】次に、第1の実施の形態によるキャリアマルチプレクサ813の回路構成の一例について、図34を用いて説明する。図34は、キャリア $C<n>$ に対応するマルチプレクサを示している( $n=0\sim8$ )。キャリア $C<n>$ 対応のマルチプレクサは、図34に示すように、論理回路50および51、ならびにインバータIV50～IV55を含む。

【0101】論理回路50は、カウンタ更新クロックCLKCNTと信号UPとを受け、信号Uを出力する。インバータIV50は、信号Uを反転した信号/Uを出力する。信号Uは、信号UPがHレベルかつカウンタ更新クロックCLKCNTがLレベルのときのみにHレベルになる。

【0102】論理回路51は、カウンタ更新クロックCLKCNTと信号DNを受け、信号Dを出力する。インバータIV51は、信号Dを反転した信号/Dを出力する。信号Dは、信号DNがHレベルかつカウンタ更新クロックCLKCNTがLレベルのときのみにHレベルになる。

【0103】インバータIV52は、信号UがHレベル（信号/UがLレベル）になると、アップキャリーUC<n>を反転して出力する。インバータIV53は、信号DがHレベル（信号/DがLレベル）になると、ダウンキャリーDC<n>を反転して出力する。

【0104】インバータIV54およびIV55は、ラッチ回路を構成する。インバータIV54およびIV55を含むラッチ回路は、インバータIV52、IV53の出力ノードと、ノードOUTとの間に配置される。ノードOUTから、キャリアC<n>が出力される。

【0105】第1の実施の形態によるDLL1000の全体構成を、図9を用いて説明する。DLL1000は、図9に示すように、クロックバッファ1、ファイン遅延素子3、コース遅延素子5、デコーダ71、グレイコードカウンタ81、位相比較器9、タイミングクロック発生器10、パルス発生器11およびレプリカ回路13を含む。

【0106】従来のDLL9000と異なる点は、バイナリカウンタ80に代わってグレイコードカウンタ81を含むこと、およびこれに伴いデコーダ70に代わってデコーダ71を含むことにある。

【0107】グレイコードカウンタ81は、カウンタ更新クロックCLKCNTに応じて、信号UP/DNに基づき、バイナリコード $ADR<0:2>$ およびグレイコード $GADR<3:8>$ を出力する。

【0108】デコーダ71は、グレイコードGADR<3:8>をデコードして、コース遅延素子5の遅延段数を決定する信号を出力する。

【0109】デコーダ71の構成の一例を、図10-図12を用いて説明する。デコーダ71は、図10-図12に示すように、信号GX<k>を出力するAND回路60#k(k=0~11)、および信号GX<k>に応じてコース遅延素子5における遅延段数を決定する信号を出力するAND回路61#0~61#63を含む。

【0110】AND回路60# $i \times 4$  ( $i=0, 1, 2$ )は、グレイコードGADR $\langle 3+i \times 2 \rangle$ を反転したグレイコード/GADR $\langle 3+i \times 2 \rangle$ と、グレイコードGADR $\langle 4+i \times 2 \rangle$ を反転したグレイコード/GADR $\langle 4+i \times 2 \rangle$ を受け、AND回路60#( $i \times 4+1$ )は、グレイコードGADR $\langle 3+i \times 2 \rangle$ とグレイコード/GADR $\langle 4+i \times 2 \rangle$ を受け  
る。

【0111】AND回路60#( $i \times 4 + 2$ )は、グレイコードGADR $\langle 3 + i \times 2 \rangle$ とグレイコードGADR $\langle 4 + i \times 2 \rangle$ とを受け、AND回路60#( $i \times 4 + 3$ )は、グレイコード/GADR $\langle 3 + i \times 2 \rangle$ とグレイコードGADR $\langle 4 + i \times 2 \rangle$ とを受ける。

【0112】AND回路61#0~61#63は、第1、第2および第3の入力ノードを有する。AND回路61#( $8 \times i$ ) および61#( $8 \times i + 7$ ) の第1入力ノードは、信号GX<0>を受け、AND回路61#( $8 \times i + 1$ ) および61#( $8 \times i + 6$ ) の第1入力ノードは、信号GX<1>を受け、AND回路61#( $8 \times i + 2$ ) および61#( $8 \times i + 5$ ) の第1入力ノードは、信号GX<2>を受け、AND回路61#( $8 \times i + 3$ ) および61#( $8 \times i + 4$ ) の第1入力ノードは、信号GX<3>を受ける( $i = 0 \sim 7$ )。

【0113】AND回路61# ( $j \times 32 + 0$ ) ~ 61# ( $j \times 32 + 3$ ) および 61# ( $j \times 32 + 28$ ) ~ 61# ( $j \times 32 + 31$ ) の第2入力ノードは、信号GX<4>を受け、AND回路61# ( $j \times 32 + 4$ ) ~ 61# ( $j \times 32 + 7$ ) および 61# ( $j \times 32 + 24$ ) ~ 61# ( $j \times 32 + 27$ ) の第2入力ノードは、信号GX<5>を受け、AND回路61# ( $j \times 32 + 8$ ) ~ 61# ( $j \times 32 + 11$ ) および 61# ( $j \times 32 + 20$ ) ~ 61# ( $j \times 32 + 23$ ) の第2入力ノードは、信号GX<6>を受け、AND回路61# ( $j \times 32 + 12$ ) ~ 61# ( $j \times 32 + 15$ ) および 61# ( $j \times 32 + 16$ ) ~ 61# ( $j \times 32 + 19$ ) の第2入力ノードは、信号GX<7>を受ける ( $j = 0, 1$ )。

【0114】AND回路61#0~61#15の第3入力ノードは、信号GX<8>を受け、AND回路61#16~61#31の第3入力ノードは、信号GX<9>を受け、AND回路61#32~61#47の第3入力ノードは、信号GX<10>を受け、AND回路61#48~61#63の第3入力ノードは、信号GX<11>を受ける。

【0115】AND回路61#m (m=0~63) の出力に応じて、コース遅延素子5の遅延段数(たとえば、m個)が決定される。

【0116】次に、第1の実施の形態によるDLL1000を有する半導体装置10000の構成の一例を、図13を用いて説明する。半導体装置10000は、図13に示すように、外部クロックCLKextを受けるDLL1000、制御信号(ロウアドレスストロブ信号/RAS、コラムアドレスストロブ信号/CAS、チップ選択信号/CS、ライトイネーブル信号/WE等)を受ける入力バッファ1001、アドレス信号An (n=0, 1, ...)を受ける入力バッファ1002、入力バッファ1001の出力を受けて内部制御信号を発生する制御信号発生回路1003、内部制御信号に応じて入力バッファ1002から出力されるロウアドレスをラッチするロウアドレスラッチ1004、内部制御信号に応じて入力バッファ1002から出力されるコラムアドレスをラッチするコラムアドレスラッチ1005、行列状に配置される複数のメモリセルと行方向に配置される複数のワード線と列方向に配置される複数のビット線とを含むメモリアレイ1006、内部制御信号に応じてロウアドレスラッチ1004の出力に基づき行選択を行うためのロウデコーダ1007、ならびに内部制御信号に応じてコラムアドレスラッチ1005の出力に基づき列選択を行うためのコラムデコーダ1008を含む。

【0117】入力バッファ1001および1002は、DLL1000の出力するクロックBUFFCLK (またはCLKin3) に同期して入力信号を取込む。クロックBUFFCLKよりもクロックCLKin3の方が、高速化により適している。

【0118】半導体装置10000はさらに、メモリアレイ1006にデータを書込み、またはメモリアレイ1006からデータを読出すためのアレイ入出力回路1009、内部制御信号に応じて読出データをラッチする読出データラッチ1010、読出データラッチ1010にラッチしたデータを取込み、データ入出力端子DQiに出力する出力バッファ1011、書込データをラッチする書込データラッチ1012、ならびにデータ入出力端子DQiからデータを取込み、書込データラッチ1012に書込データを出力する入力バッファ1013を含む。

【0119】出力バッファ1011は、DLL1000の出力する内部クロックCLKin2に同期して動作する。

【0120】第1の実施の形態によるグレイコードカウンタを含むDLLにより得られる効果について述べる。たとえば、図1, 2の例(4ビットグレイコード)で説明する。上記したように、グレイコードカウンタにおいては、「アップ」においても「ダウン」においても、1ビットだけキャリーが立つ。

【0121】そこで、現在のレジスタ値がデシマルコード“6”=グレイコード“0101”であったとする。信号UPがHならば、キャリーC<0>=“1”、キャリーC<1>~C<3>=“0”である。一方、信号DN=Hならば、キャリーC<1>=“1”、キャリーC<0>, <2>, <3>=“0”である。ここで、位相比較器9においてメタステーブル状態が発生し、信号UPおよびDNが不定値となってしまったとする。この場合においてキャリー値がどのようになるかを考察する。

【0122】キャリーC<2>, <3>は、信号UPまたは信号DNのいずれがHであっても“0”である。したがって、信号UPおよびDNが不定になってもキャリーC<2>~C<3>は不定にならず“0”を保持する。

【0123】キャリーC<0>, C<1>は、信号UPおよびDNの値により“0”または“1”の値になる。したがって、信号UPおよびDNが不定になるとキャリーC<0>, C<1>も不定になる。つまり、これによる次の4通りの結果がいずれかの確率で発生してしまう。

【0124】(イ) キャリーC<0>=“0”、キャリーC<1>=“0”のときすべてのキャリーが“0”になるので、次のレジスタ値はデシマルコード6=グレイコード0101のままである。

【0125】(ロ) キャリーC<0>=“1”、キャリーC<1>=“0”のとき次のレジスタ値はデシマル7=グレイコード0100である。

【0126】(ハ) キャリーC<0>=“0”、キャリーC<1>=“1”のとき次のレジスタ値はデシマルコード5=グレイコード1111である。

【0127】(ニ) キャリーC<0>=“1”、キャリーC<1>=“1”のとき、次のレジスタ値はデシマル4=グレイコード0110である。

【0128】この例に表わされるように、メタステーブル状態が発生した後は、元の状態“6”から遷移する可能性があるのは“4”、“5”、“6”、“7”のみである。すなわち、元の状態から最大でも“2”だけ離れた値にしか遷移しない。

【0129】ここでは、元の状態が“6”である場合について説明したが、元の状態が“6”以外の値であっても「元の状態から最大でも“2”だけ離れたところにしか遷移しない」ことがわかる。

【0130】また、この事情は4ビットグレイコードカウンタのみならず、一般のnビットグレイコードカウンタについても言える。

【0131】したがって、グレイコードカウンタを用いることにより、メタステーブル状態が発生しても最大でも元のレジスタ値から2はなれたところまでしかレジスタ値が飛ばない。従来例であるバイナリカウンタでは、元のレジスタ値から全く予想ができないところにレジス

タ値が飛んでしまう可能性があったことを考えると、本発明の示す効果が大きいことがわかる。

【0132】このため、グレイコードカウンタを含むDLLにおいては、DLLの遅延値が安定化され、遅延のとりが最小化されるので、従来例で見たような不良が起こる可能性がなくなるという効果がある。

【0133】[第2の実施の形態]第2の実施の形態においては、DDR・SDRAMに搭載するDLLについて説明する。DDR・SDRAM用のDLL2000について、図14を用いて説明する。DLL2000には、図14に示すように、外部からクロックCLKextおよびZCLKextが入力される。クロックZCLKextはクロックCLKextと位相が180度異なる。

【0134】DLL2000は、クロックバッファ1、2、ファイン遅延素子3、4、コース遅延素子5、6、デコーダ71、グレイコードカウンタ81、位相比較器9、タイミングクロック発生器10、パルス発生器11および12、ならびにレプリカ回路13を含む。

【0135】ファイン遅延素子4は、ファイン遅延素子3と、コース遅延素子6は、コース遅延素子5とそれぞれ同じ構成を有する。また、パルス発生器12は、パルス発生器11と同じ構成を有する。

【0136】クロックバッファ1および2にはともに、外部クロックCLKext、ZCLKextが入力されている。クロックバッファ1、2は、図31、32に示すように、トランジスタTr1、Tr2、Tr3、Tr4およびインバータIV10を含む。クロックバッファ1は、クロックBUFFCLKを、クロックバッファ2は、クロックBUFFZCLKを出力する。

【0137】位相比較器9は、レプリカ回路13の出力するクロックFBCLKとクロックBUFFCLKとの位相を比較して、信号UPF/DNFを出力する。グレイコードカウンタ81から出力されるバイナリコードADR<0:2>は、ファイン遅延素子3、4に入力される。

【0138】デコーダ71は、グレイコードカウンタ81の出力するグレイコードGADR<3:8>をデコードして、コース遅延素子5、6に出力する。

【0139】クロックBUFFZCLKは、ファイン遅延素子4とコース遅延素子6とを通過した後、パルス発生器12に入力される。パルス発生器12から、クロックZCLKin2が出力される。内部クロックZCLKin2は、内部クロックCLKin2とともにDDR・SDRAMのデータ出力に用いられる。

【0140】ここで、DLL2000の動作について、図15を用いて説明する。図15を参照して、信号BUFFCLKは、クロックバッファ1によりクロックCLKextの立上がりとクロックZCLKextの立下がりとのクロックポイントにおいて発生する。一方、クロ

ックBUFFZCLKは、クロックバッファ2により、クロックZCLKextの立上がりとクロックCLKextの立下がりのクロスポイントとにおいて発生される。

【0141】内部クロックCLKin2は、外部クロックCLKextの立上がりと外部クロックZCLKextの立下がりとのクロスポイントに同期してデータを入力する際に用い、内部クロックZCLKin2は、外部クロックZCLKextの立上がりと内部クロックCLKextの立下がりとのクロスポイントに同期してデータを入力する際に用いる。

【0142】ここで、第2の実施の形態によるDLL2000を有する半導体装置20000の構成の一例を、図16を用いて説明する。半導体装置20000は、図16に示すように、外部クロックCLKextおよびZCLKextを受けるDLL2000、制御信号(ロウアドレスストロブ信号/RAS、コラムアドレスストロブ信号/CAS、チップ選択信号/CS、ライトイネーブル信号/WE等)を受ける入力バッファ2001、アドレス信号An(n=0,1,...,11)およびバンクアドレス信号BA0,BA1を受ける入力バッファ2002、入力バッファ2001の出力を受けて内部制御信号を発生する制御信号発生回路2003、内部制御信号に応じて入力バッファ1002から出力されるロウアドレスをラッチするロウアドレスラッチ2004、内部制御信号に応じて入力バッファ2002から出力されるコラムアドレスをラッチするコラムアドレスラッチ2005、行列状に配置される複数のメモリセルと行方向に配置される複数のワード線と列方向に配置される複数のビット線とを含むメモリアレイ2006、内部制御信号に応じてロウアドレスラッチ2004の出力に基づき行選択を行うためのロウデコーダ2007、ならびに内部制御信号に応じてコラムアドレスラッチ2005の出力に基づき列選択を行うためのコラムデコーダ2008を含む。入力バッファ2001および2002は、DLL2000の出力するクロックBUFFCLK(またはCLKin3)に同期して入力信号を取込む。クロックBUFFCLKよりもクロックCLKin3の方が、高速化により適している。

【0143】半導体装置20000はさらに、メモリアレイ2006にデータを書込みまたはメモリアレイ2006からデータを読み出すためのアレイ入出力回路2009、内部制御信号に応じて読出データをラッチする読出データラッチ2010、読出データラッチ2010にラッチしたデータを取込み、データ入出力端子DQiに出力する出力バッファ2011、書込データをラッチする書込データラッチ2012、ならびにデータ入出力端子DQiからデータを取込み、書込データラッチ2012に書込データを出力する入力バッファ2013を含む。

【0144】出力バッファ2011は、DLL2000

の出力する内部クロックCLK<sub>in2</sub>およびZCLK<sub>in2</sub>に同期して動作する。したがって、半導体装置20000(DDR・SDRAM)は、1クロック周期中に2回のデータを出力することができる。

【0145】今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

【0146】

【発明の効果】このように、本発明によるDLL回路は、遅延量を調整する遅延レジスタとしてグレイコードカウンタを用いる。これにより、キャリーが1ビットしか立たないので、DLLの遅延値が安定化され、遅延のとびが最小化される。これにより、安定した高速動作を実現することが可能となる。

【0147】また、本発明によるDLL回路を含む半導体装置によれば、外部クロックを遅延して内部クロックを発生する際に、遅延量を調整する遅延レジスタとしてグレイコードカウンタを用いる。これにより、不連続な遅延の「飛び」を原理的に最小化することが可能となる。したがって、入力バッファや出力バッファを当該内部クロックで制御することで、安定した高速動作を実現することが可能となる。

【0148】また、本発明によるクロック同期により動作するシステムのための制御方法によれば、グレイコードを用いて遅延量を制御するため、不連続な遅延の「飛び」を原理的に最小化することが可能となる。これにより、データを安定したタイミングで入力することができる。同じく、データを安定したタイミングで出力することができる。

【図面の簡単な説明】

【図1】 4ビットのグレイコードについて説明するための図である。

【図2】 4ビットのグレイコードを4ビットのバイナリコードに変換する変換式について説明するための図である。

【図3】 nビットのグレイコードをnビットのバイナリコードに変換する変換式について説明するための図である。

【図4】 グレイコードにおけるキャリーについて説明するための図である。

【図5】 第1の実施の形態によるグレイコードカウンタ(9ビット)の構成の一例を示す図である。

【図6】 キャリーC<0>〜C<8>の発生過程を説明するためのタイミングチャートである。

【図7】 第1の実施の形態によるグレイコードレジスタ810の回路構成の一例を示す図である。

【図8】 第1の実施の形態によるバイナリコード発生

器811の回路構成の一例を示す図である。

【図9】 第1の実施の形態によるDLL1000の全体構成を示す図である。

【図10】 第1の実施の形態によるデコーダ71の構成要素を示す図である。

【図11】 第1の実施の形態によるデコーダ71の構成要素を示す図である。

【図12】 第1の実施の形態によるデコーダ71の構成要素を示す図である。

【図13】 第1の実施の形態によるDLL1000を有する半導体装置10000の構成の一例を示す図である。

【図14】 第2の実施の形態によるDLL2000の全体構成を示す図である。

【図15】 第2の実施の形態によるDLL2000の動作について説明するためのタイミングチャートである。

【図16】 第2の実施の形態によるDLL2000を有する半導体装置20000の構成の一例を示す図である。

【図17】 データの出力タイミングについて説明するためのタイミングチャートである。

【図18】 データの出力タイミングについて説明するためのタイミングチャートである。

【図19】 システムクロック周期Tが短い場合における問題点を説明するためのタイミングチャートである。

【図20】 DDR・SDRAMにおける動作タイミングを説明するためのタイミングチャートである。

【図21】 DLLを用いた部品における動作を説明するためのタイミングチャートである。

【図22】 DLLを用いない場合のデータ入力における動作について説明するためのタイミングチャートである。

【図23】 ラッチ回路910の構成の一例を示す図である。

【図24】 DLLを用いた場合のデータ入力における動作について説明するためのタイミングチャートである。

【図25】 従来のDLL9000の構成の一例を示す図である。

【図26】 DLL9000の動作について説明するためのタイミングチャートである。

【図27】 DLL9000の動作について説明するためのタイミングチャートである。

【図28】 (a), (b)は、信号UPF, DNFの値について説明するためのタイミングチャートである。

【図29】 デコーダ70の構成の一例を示す図である。

【図30】 位相比較器9の構成の一例を示す図である。

【図31】 クロックバッファ1の構成の一例を示す図である。

【図32】 クロックバッファ2の構成の一例を示す図である。

【図33】 第1の実施の形態によるアップ・ダウンキャリ発生器812の回路構成の一例を示す図である。

【図34】 第1の実施の形態によるキャリマルチプレクサ813の回路構成の一例を示す図である。

【符号の説明】

1, 2 クロックバッファ、3, 4 ファイン遅延素子、5, 6 コース遅延素子、9 位相比較器、10 タイミングクロック発生器、11, 12 パルス発生器、13 レプリカ回路、20 ゲート、40 バッファ、71 デコーダ、81 グレイコードカウンタ、8

10 グレイコードレジスタ、811 バイナリコード発生器、812 アップ・ダウンキャリ発生器、813 キャリマルチプレクサ、1000, 2000 D LL、1001, 1002, 1013, 2001, 2002, 2013 入力バッファ、1003, 2003 制御信号発生回路、1004, 2004 ロウアドレスラッチ、1005, 2005コラムアドレスラッチ、1006, 2006 メモリアレイ、1007, 2007 ロウデコーダ、1008, 2008 コラムデコーダ、1009, 2009アレイ入出力回路、1010, 2010 読出データラッチ、1011, 2011 出力バッファ、1012, 2012 書込データラッチ、10000, 20000 半導体装置。

【図1】

4ビットグレイコードの図解

デシマルコード	バイナリコード ADR<3~0>	グレイコード GADR<3~0>	・キャリ
0	0 0 0 0	0 0 0 0	
1	0 0 0 1	0 0 0 1	
2	0 0 1 0	0 0 1 1	
3	0 0 1 1	0 0 1 0	
4	0 1 0 0	0 1 1 0	
5	0 1 0 1	0 1 1 1	
6	0 1 1 0	0 1 0 1	
7	0 1 1 1	0 1 0 0	
8	1 0 0 0	1 1 0 0	
9	1 0 0 1	1 1 0 1	
10	1 0 1 0	1 1 1 1	
11	1 0 1 1	1 1 1 0	
12	1 1 0 0	1 0 1 0	
13	1 1 0 1	1 0 1 1	
14	1 1 1 0	1 0 0 1	
15	1 1 1 1	1 0 0 0	

↓ ダウン方向  
↑ アップ方向

【図3】

nビットグレイコードをnビットバイナリコードに変換する変換式

バイナリコード ADR<n-1~0>	グレイコード GADR<n-1~0>
ADR<n-1>	= GADR<n-1>
ADR<n-2>	= Exor(GADR<n-1>, GADR<n-2>)
ADR<n-3>	= Exor(GADR<n-1>~GADR<n-3>)
...	...
ADR<0>	= Exor(GADR<n-1>~GADR<0>)

【図2】

4ビットグレイコードを4ビットバイナリコードに変換する変換式

バイナリコード ADR<3~0>	グレイコード GADR<3~0>
ADR<3>	= GADR<3>
ADR<2>	= Exor(GADR<3>, GADR<2>)
ADR<1>	= Exor(GADR<3>~GADR<1>)
ADR<0>	= Exor(GADR<3>~GADR<0>)

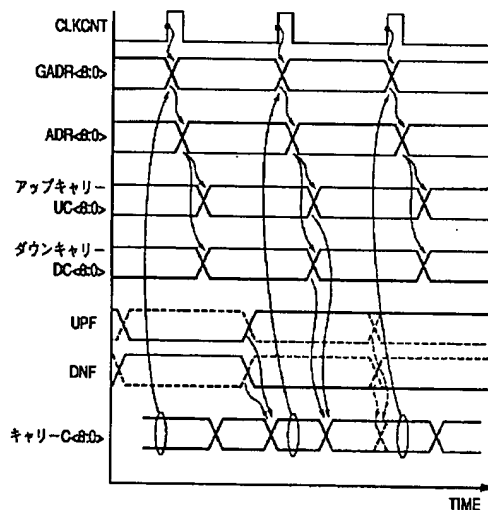
【図4】

4ビットグレイコードのキャリの場所の例示

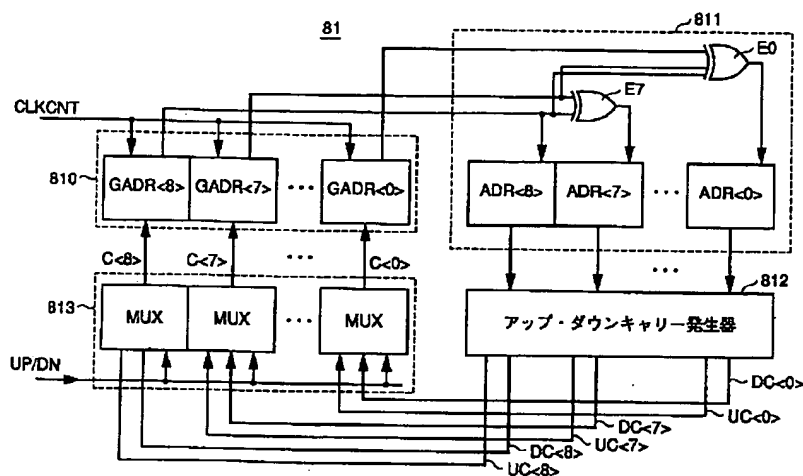
デシマルコード	バイナリコード	グレイコード
5	0 1 0 1	0 1 1 1
8	0 1 ① ②	0 1 0 1
7	0 1 1 1	0 1 0 0

↓ ダウン  
↑ アップ

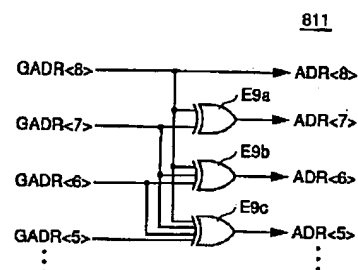
【図6】



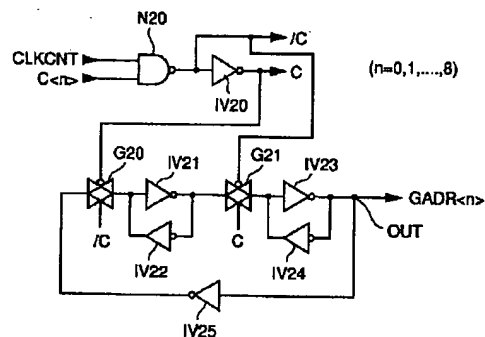
【図5】



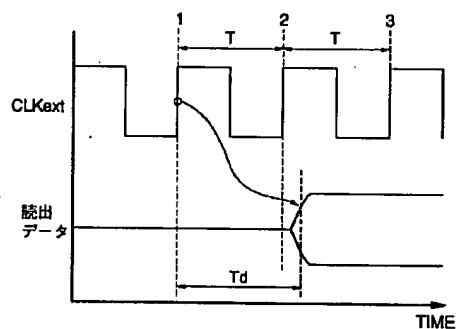
【図8】



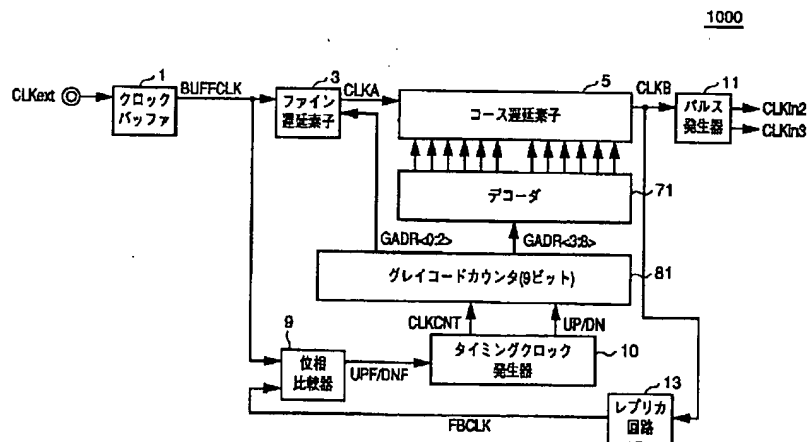
【図7】



【図19】

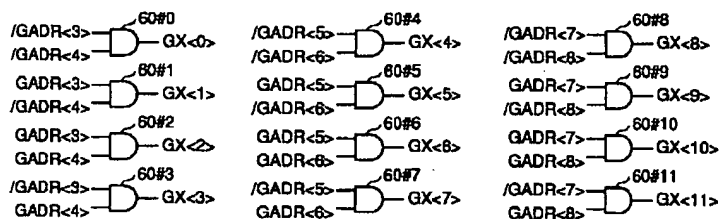


【図9】

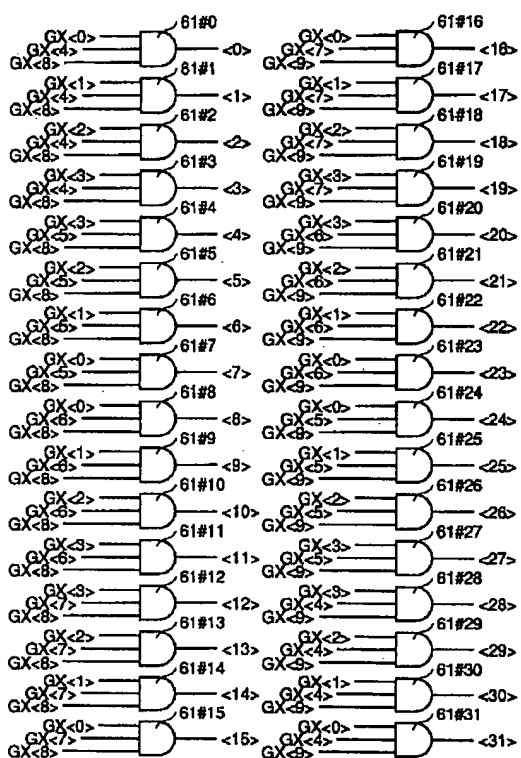




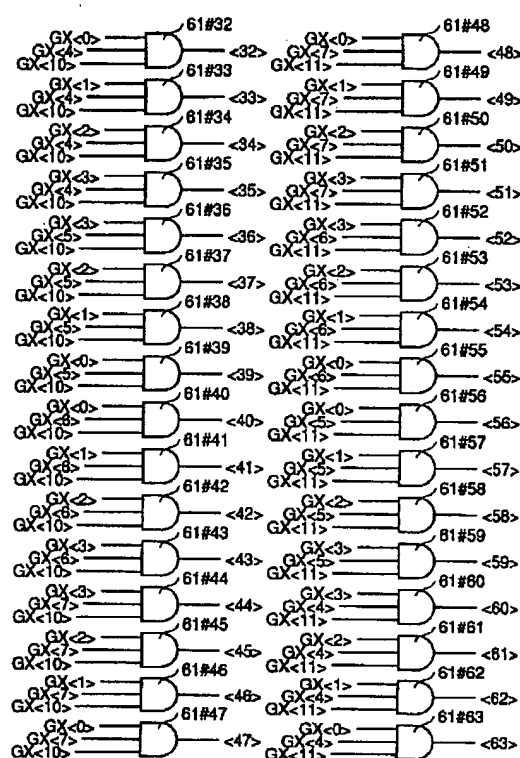
【図10】



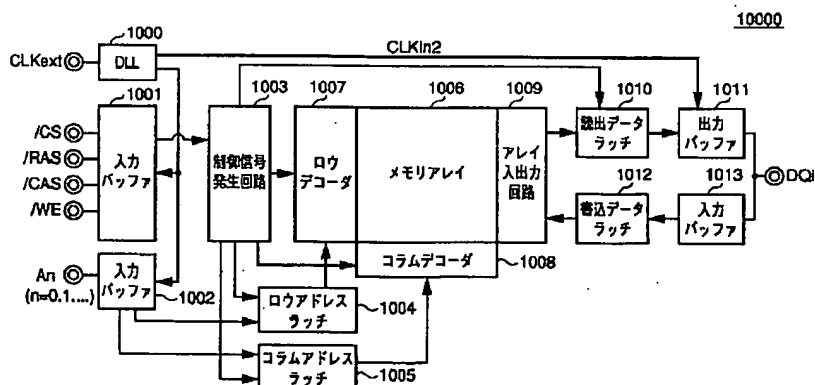
【図11】



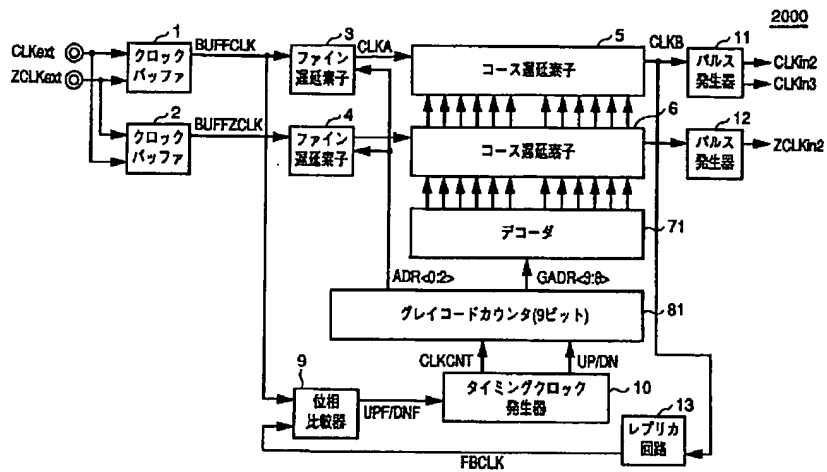
【図12】



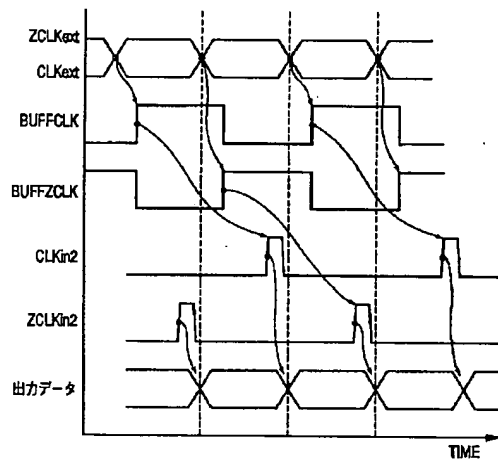
【図13】



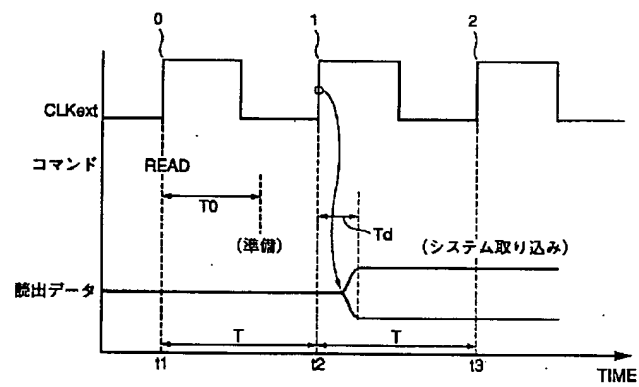
【図14】



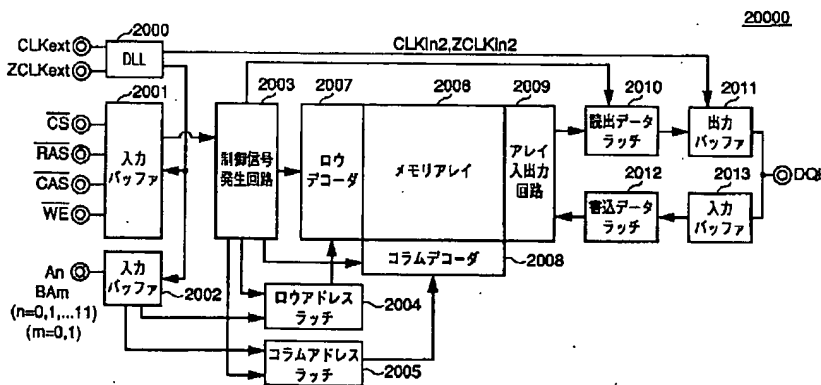
【図15】



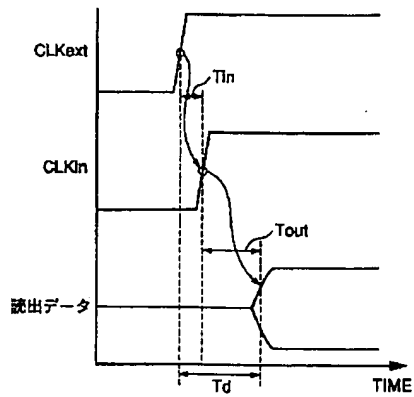
【図17】



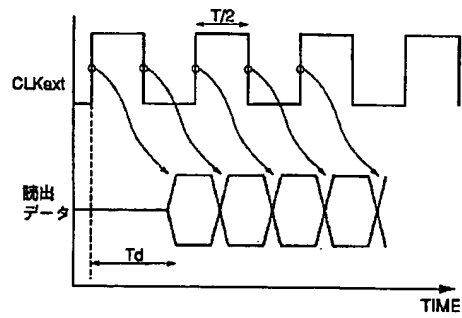
【図16】



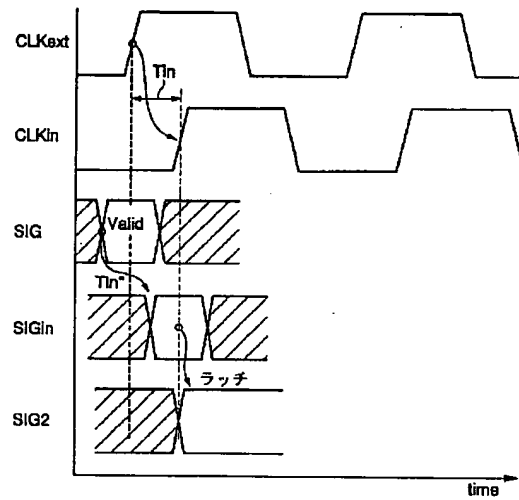
【図18】



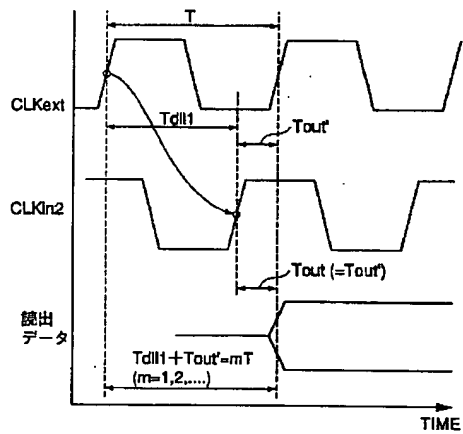
【図20】



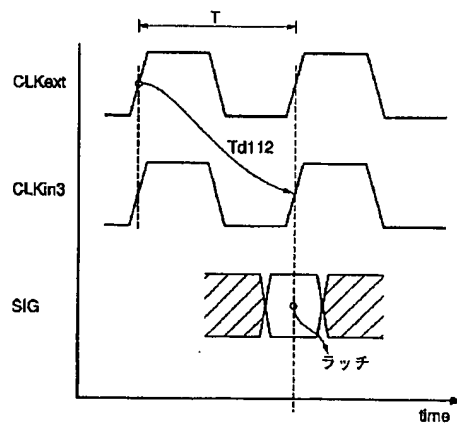
【図22】



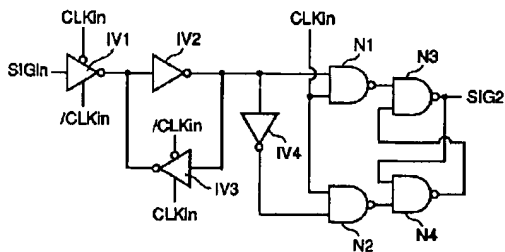
【図21】



【図24】

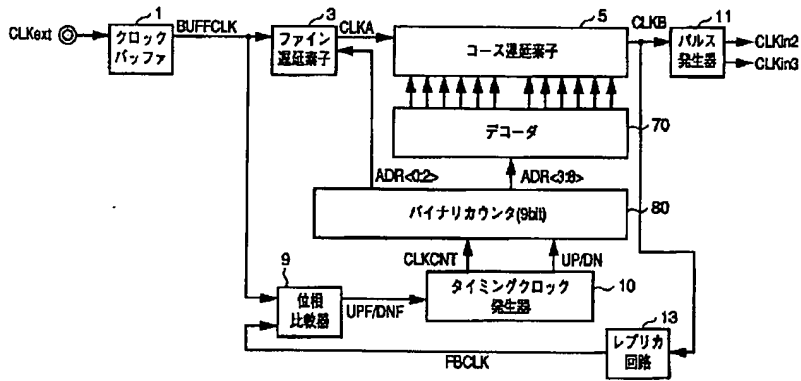


【図23】

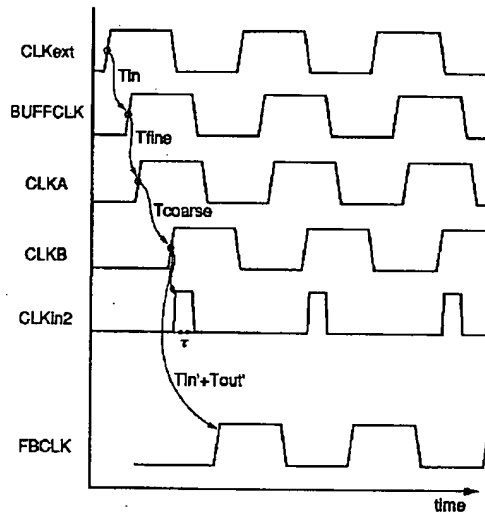


【図25】

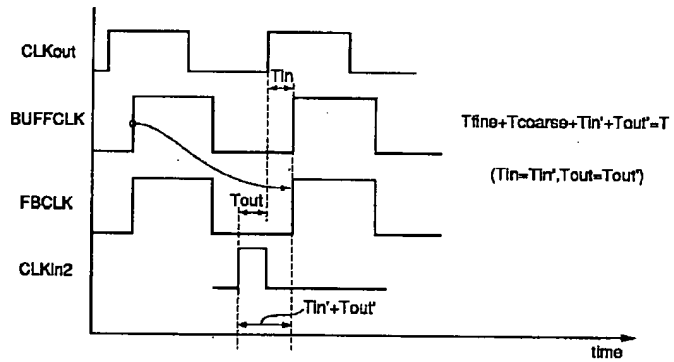
9000



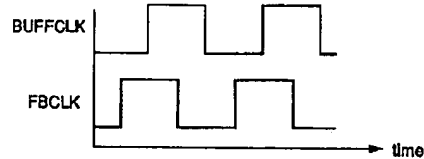
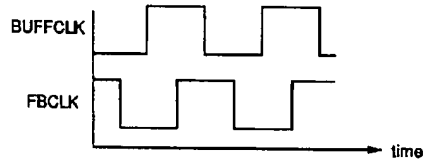
【図26】



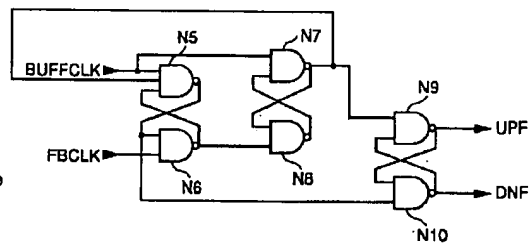
【図27】



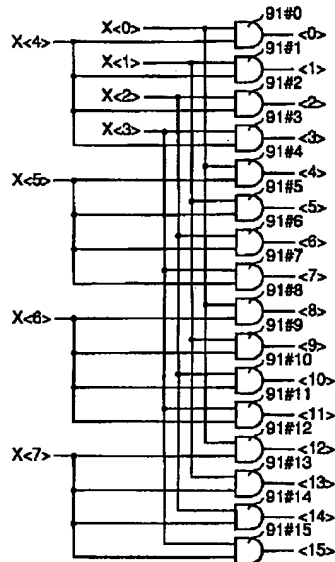
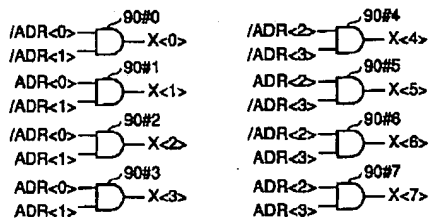
【図28】

(a) アップ( $UPF="H", DNF="L"$ )(b) ダウン( $DNF="H", UPF="L"$ )

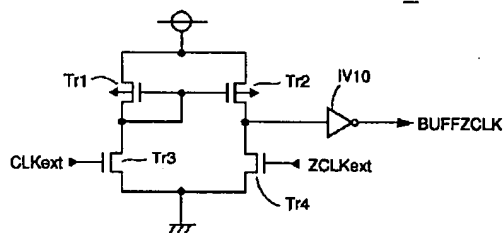
【図30】



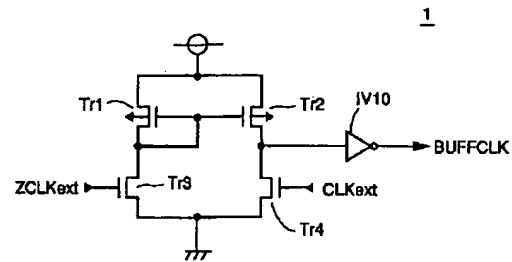
【図29】



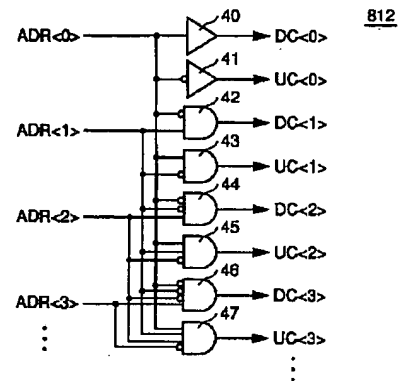
【図32】



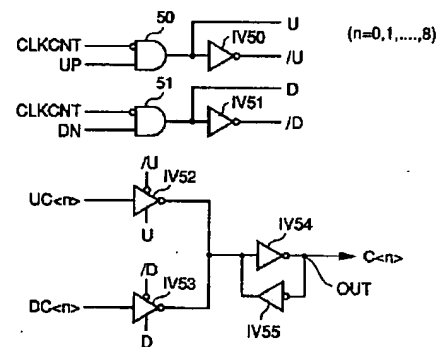
【図31】



【図33】



【図34】



フロントページの続き

(51) Int. Cl.<sup>7</sup>

H03K 23/64

H03M 7/16

識別記号

FI

G11C 11/34

H03K 5/00

メモード (参考)

354C

362S

V